

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **11203372 A**

(43) Date of publication of application: **30.07.99**

(51) Int. Cl.

G06F 19/00
B65G 1/137

(21) Application number: **10002589**

(22) Date of filing: **08.01.98**

(71) Applicant: **FUJITSU LTD**

(72) Inventor: **UEHARA TADAHIRO**
YOSHIDA HIROYUKI
YAMAMOTO RIEKO
SAKURAI HIROSHI

**(54) METHOD FOR MANAGING STOCK FOR
AUTOMATIC ASSIGNING STOCK AND DEVICE
THEREFOR**

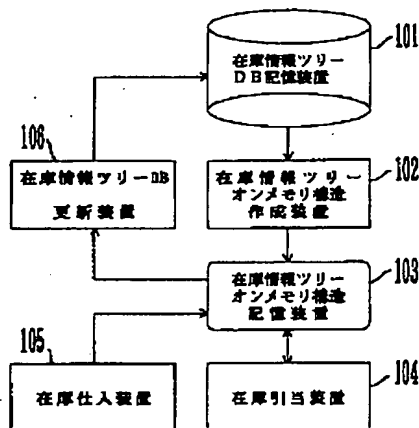
hierarchy in which the attribute which is not designated
in the assignment condition is defined.

COPYRIGHT: (C)1999,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To realize automatic stock assignment based on a preliminarily decided policy even when condition designation is vague at the time of stock assignment.

SOLUTION: Each attribute of merchandise is assigned to each hierarchy of a stock information tree on a stock information tree-on memory structure storage device 103. At least the kind of the attribute of merchandise, the value of the attribute, the number of stock of merchandise fulfilling a classification condition expressed with the position of the node in the tree, and link relation with another node are defined in the node constituting the stock information tree. A stock assigning device 104 retrieves each node on the stock information tree according to an assignment condition designated by the combination of the attribute of the merchandise with the value, and executes stock assignment for updating the number of stock of each node. At the time of retrieving each node, each node is retrieved in a priority order corresponding to the position relation of each node in the hierarchy for a



(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-203372

(43)公開日 平成11年(1999) 7月30日

(51)Int.Cl.⁹

G 0 6 F 19/00

B 6 5 G 1/137

識別記号

F I

G 0 6 F 15/24

B 6 5 G 1/137

A

審査請求 未請求 請求項の数 8 O L (全 21 頁)

(21)出願番号 特願平10-2589

(22)出願日 平成10年(1998) 1月 8日

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(72)発明者 上原 忠弘

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(72)発明者 吉田 裕之

神奈川県川崎市中原区上小田中4丁目1番
1号 富士通株式会社内

(74)代理人 弁理士 大曾 義之 (外1名)

最終頁に続く

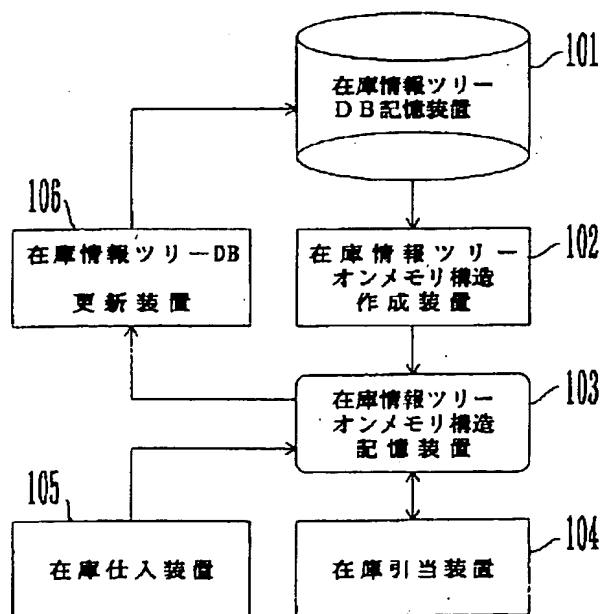
(54)【発明の名称】 自動在庫引当が可能な在庫管理方法及び装置

(57)【要約】

【課題】 在庫引当時に条件指定が曖昧であっても、予め決められた方針に沿った自動的な在庫引当を実現することにある。

【解決手段】 在庫情報ツリーオンメモリ構造記憶装置103上の在庫情報ツリーの各階層には商品の各属性が割り当てられる。その在庫情報ツリーを構成するノードには、商品の属性の種類と、その属性の値と、そのノードのツリー内での位置によって表わされる分類条件を満たす商品の在庫数と、他のノードとのリンク関係とが少なくとも定義される。在庫引当装置104は、商品の属性とその値の組合せによって指定される引当条件に従って在庫情報ツリー上の各ノードを検索しながら、各ノードの在庫数を更新する在庫引当を実行する。各ノードの検索時に、引当条件に指定されていない属性が定義されている階層に対しては、その階層内での各ノードの位置関係に対応する優先順位で各ノードが検索される。

本発明の実施の形態のシステム構成図



【特許請求の範囲】

【請求項1】 管理対象に対して在庫管理を行う在庫管理方法であって、

各階層に前記管理対象の各属性が割り当てられる2進木データを構成するノードに、前記管理対象の属性の種類と、該属性の値と、そのノードの前記2進木データ上での位置によって表わされる分類条件を満たす前記管理対象の在庫数と、他のノードとのリンク関係とを少なくとも定義し、

在庫引当時に引当条件として前記管理対象の属性とその値とを指定し、

該引当条件に従って前記2進木データ上の前記各ノードを検索しながら、該各ノードの在庫数を更新する在庫引当を実行し、

前記各ノードの検索時に、前記引当条件に指定されていない属性が定義されている階層に対しては、該階層内での前記各ノードの位置関係に対応する優先順位で該各ノードを検索する、

過程を含むことを特徴とする自動在庫引当が可能な在庫管理方法。

【請求項2】 管理対象に対して在庫管理を行う在庫管理装置であって、

前記管理対象の属性の種類と、該属性の値と、各階層に前記管理対象の各属性が割り当てられる2進木データ上での位置によって表わされる分類条件を満たす前記管理対象の在庫数と、他のノードとのリンク関係とが少なくとも定義され、前記2進木データを構成するノードが割り当てられ、前記属性の種類と該属性の値と前記在庫数と前記リンク関係にそれぞれ対応するフィールドを有するレコードによって構成されるデータベースと、

前記管理対象の属性とその値の組合せによって記述される引当条件に従って、前記データベースを介して、前記2進木データ上の前記各ノードを検索しながら、該各ノードの在庫数を更新する在庫引当を実行し、前記各ノードの検索時に、前記引当条件に指定されていない属性が定義されている階層に対しては、該階層内での前記各ノードの位置関係に対応する優先順位で該各ノードを検索する在庫引当装置と、

を含むことを特徴とする自動在庫引当が可能な在庫管理装置。

【請求項3】 請求項2に記載の装置であって、前記管理対象の属性とその値の組合せによって記述される仕入条件に従って、前記データベースを介して、前記2進木データ上の前記各ノードを検索しながら、該各ノードの在庫数を更新する在庫仕入を実行する在庫仕入装置を更に含む、

ことを特徴とする自動在庫引当が可能な在庫管理装置。

【請求項4】 請求項2又は3の何れか1項に記載の装置であって、

前記データベーステーブルからメモリ装置上に前記2進木データを展開する2進木データオンメモリ構造作成装置を更に含む、

前記在庫引当装置又は前記在庫仕入装置は、前記メモリ装置上に展開された2進木データに対して前記在庫引当又は在庫仕入を実行する、

ことを特徴とする自動在庫引当が可能な在庫管理装置。

【請求項5】 請求項4に記載の装置であって、前記レコードは、前記2進木データ上での前記ノードの位置関係に応じた値を有する識別情報を格納する識別情報フィールドを更に含む、

前記2進木データオンメモリ構造作成装置は、該識別情報フィールドの内容に従って、前記データベーステーブルからメモリ装置上に前記2進木データを展開する、ことを特徴とする自動在庫引当が可能な在庫管理装置。

【請求項6】 請求項4に記載の装置であって、前記レコードは、前記管理対象自体の種別を格納する管理対象種別フィールドを更に含む、

前記2進木データオンメモリ構造作成装置は、該管理対象種別フィールドの内容に従って、前記データベーステーブルからメモリ装置上に前記管理対象種別毎の前記2進木データを展開する、

ことを特徴とする自動在庫引当が可能な在庫管理装置。

【請求項7】 請求項4に記載の装置であって、前記メモリ装置上に展開された2進木データを構成する各ノードは、該各ノードに対応する在庫数が前記在庫引当装置又は前記在庫仕入装置によって更新されたか否かを示すノード更新情報を保持し、

該ノード更新情報に従い、メモリ装置上に展開された2進木データの更新状態を前記データベーステーブルに反映させるデータベース更新装置を更に含む、

ことを特徴とする自動在庫引当が可能な在庫管理装置。

【請求項8】 コンピュータにより使用されたときにそれによって読み出されるプログラムを記録した記録媒体であって、

各階層に在庫管理の管理対象の各属性が割り当てられる2進木データを構成するノードに、前記管理対象の属性の種類と、該属性の値と、そのノードの前記2進木データ上での位置によって表わされる分類条件を満たす前記管理対象の在庫数と、他のノードとのリンク関係とを少なくとも定義する機能と、

在庫引当時に引当条件として前記管理対象の属性とその値とを指定する機能と、

該引当条件に従って前記2進木データ上の前記各ノードを検索しながら、該各ノードの在庫数を更新する在庫引当を実行する機能と、

前記各ノードの検索時に、前記引当条件に指定されていない属性が定義されている階層に対しては、該階層内での前記各ノードの位置関係に対応する優先順位で該各ノードを検索する機能と、

を前記コンピュータに行わせるためのプログラムを記録したコンピュータ読出し可能記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、販売物流業務における在庫引当処理に関する。

【0002】

【従来の技術及び発明が解決しようとする課題】販売物流業務における商品の在庫管理技術として、データベースシステムを用いて行う技術が普及している。

【0003】このような在庫管理を目的とするデータベース上のテーブルにおいては、一般に、1つの商品が1つのレコードエントリとして登録され、各レコードは、その商品の色、サイズ等の属性を示す複数の同一のフィールドから構成される。

【0004】このようなデータベース上のテーブルに対して、特定の条件を満足する商品を抽出するための在庫引当が行われる場合、従来は、そのテーブル上の属性（フィールド）毎に問合せ条件を指定してデータベース検索が行われていた。

【0005】そして、このような従来の在庫引当処理においては、条件が指定されている属性は在庫引当の結果に影響を及ぼすが、特に条件が指定されない属性が在庫引当の結果に影響を及ぼすことはなかった。

【0006】従って、従来は、在庫引当の方針は全て、所定の属性に対する条件の指定という形式で具現化する必要があったため、条件の設定にある程度の経験が必要であり、必ずしも柔軟な在庫引当処理ができていたとは言い難かった。

【0007】具体的には、従来、例えば、Lサイズの商品よりもMサイズの商品が優先して引き当てられるような在庫引当を行うためには、まず、サイズ属性に対してMサイズという条件を指定して在庫引当を行った後に、サイズ属性に対してLサイズという条件を指定して在庫引当を行うという複数段階の在庫引当処理が必要であった。従って、従来は、例えば、在庫引当時にサイズ指定がなされなかった場合にLサイズの商品よりもMサイズの商品が優先して引き当てられる、というような自動的な在庫引当のための方針をデータベースに対して設定することは困難であった。

【0008】本発明の課題は、在庫引当時に条件指定が曖昧であっても、予め決められた方針に沿った自動的な在庫引当を実現することにある。

【0009】

【課題を解決するための手段】本発明は、管理対象に対して在庫管理を行う在庫管理装置を前提とする。そしてまず、管理対象の属性の種類と、その属性の値と、各階層に管理対象の各属性が割り当てられる2進木データ上での位置によって表わされる分類条件を満たす管理対象の在庫数と、他のノードとのリンク関係とが少なくとも

定義され、2進木データを構成するノードが割り当てられ、属性の種類とその属性の値と在庫数とリンク関係にそれぞれ対応するフィールドを有するレコードによって構成されるデータベーステーブル（在庫情報ツリーDB）を有する。

【0010】次に、管理対象の属性とその値の組合せによって記述される引当条件に従って、データベーステーブルを介して、2進木データ上の各ノードを検索しながら、その各ノードの在庫数を更新する在庫引当を実行し、各ノードの検索時に、引当条件に指定されていない属性が定義されている階層に対しては、その階層内での各ノードの位置関係に対応する優先順位でその各ノードを検索する在庫引当装置（104）を有する。

【0011】上述の発明の構成により、在庫引当時に条件指定が曖昧であっても、2進木データの木構造における検索の優先順位に対応する在庫引当戦略方針に沿った自動的な在庫引当が実現される。

【0012】上述の発明の構成において、管理対象の属性とその値の組合せによって記述される仕入条件に従って、データベーステーブルを介して、2進木データ上の各ノードを検索しながら、その各ノードの在庫数を更新する在庫仕入を実行する在庫仕入装置（105）を更に含むように構成することができる。

【0013】また、上述の発明の構成において、データベーステーブルからメモリ装置上に2進木データを展開する2進木データオンメモリ構造作成装置（在庫情報ツリーオンメモリ構造作成装置102）を更に含み、前述の在庫引当装置又は在庫仕入装置は、メモリ装置上に展開された2進木データに対して在庫引当又は在庫仕入を実行するように構成することができる。

【0014】この発明の構成により、2進木データへの高速アクセスが可能となる。また、上述の発明の構成において、データベース上の各レコードは、2進木データ上でのノードの位置関係に応じた値を有する識別情報を格納する識別情報フィールドを更に含み、2進木データオンメモリ構造作成装置は、その識別情報フィールドの内容に従って、データベーステーブルからメモリ装置上に2進木データを展開するように構成することができる。

【0015】この発明の構成により、データベーステーブルに対する簡単な問合せ文によって、データベーステーブルからメモリ装置上に2進木データを展開することができる。

【0016】また、上述の発明の構成において、データベース上の各レコードは、管理対象自体の種別を格納する管理対象種別フィールドを更に含み、2進木データオンメモリ構造作成装置は、その管理対象種別フィールドの内容に従って、データベーステーブルからメモリ装置上に管理対象種別毎の2進木データを展開するように構成することができる。

【0017】この発明の構成でもやはり、データベーステーブルに対する簡単な問合せ文によって、データベーステーブルからメモリ装置上に2進木データを展開することができる。

【0018】更に、上述の発明の構成において、メモリ装置上に展開された2進木データを構成する各ノードは、その各ノードに対応する在庫数が在庫引当装置又は在庫仕入装置によって更新されたか否かを示すノード更新情報を保持し、そのノード更新情報に従い、メモリ装置上に展開された2進木データの更新状態をデータベーステーブルに反映させるデータベース更新装置（在庫情報ツリーDB更新装置106）を更に含むように構成することができる。

【0019】この発明の構成により、メモリ装置上の2進木データとデータベーステーブルとの効率的な整合化を図ることができる。また、本発明は、コンピュータにより使用されたときに、上述の本発明の第1の態様の構成によって実現される機能と同様の機能をコンピュータに行わせるためのコンピュータ読出し可能記録媒体として構成することもできる。

【0020】

【発明の実施の形態】以下、図面を参照しながら本発明の実施の形態について詳細に説明する。

本発明の実施の形態の構成及び基本原理

図1は、本発明の実施の形態の構成図である。

【0021】本発明の実施の形態では、在庫情報がツリー（2進木）構造を有するデータベースによって管理されることが特徴であり、全ての在庫情報に関するツリー構造は、ハードディスク等の補助記憶装置によって構成される図1に示される在庫情報ツリーデータベース（在庫情報ツリーDB）記憶装置101に保持される。

【0022】この場合、商品の1つの属性に在庫情報ツリーの1つの階層が割り当てられ、その階層に、その属性が取り得る各値に対応する各ノードが配置される。在庫情報ツリーDB記憶装置101に記憶される在庫情報ツリーDBのテーブル上では、図2に示されるように、1つのノードは1つのレコードを構成する。各レコードは、図2に示されるように、各ノードを識別するためのノードIDを格納するフィールド、属性の種類（色、サイズ等）が格納されるフィールド、属性の値（赤、青、Mサイズ、Lサイズ等）が格納されるフィールド、そのレコードに対応するノードのツリー構造上の位置によって表わされる分類条件を満たす商品の在庫数が格納されるフィールド、及びツリー構造を表現するためにそのレコードに対応するノードにリンクする親ノードのノードIDを格納するフィールド（親IDフィールド）等を有する。

【0023】図2に示されるデータ構造を有する在庫情報ツリーDBにより、商品毎に、図3(a)及び(b)に示される在庫情報ツリー構造を表現することが可能とな

る。図3において、矩形で囲まれた各部分がノードに対応し、矩形間の直線が在庫情報ツリーDB上の親IDフィールドに格納される情報で表現されるノード間のリンクを表わしている。また“商品”、“色”、“サイズ”等の別は在庫情報ツリーDB上の属性フィールドに格納される情報に対応し、“商品A”、“商品B”、“赤”、“青”、“黄”、“M”、“L”、“S”、等の別は在庫情報ツリーDB上の値フィールドに格納される情報に対応し、“800”、“600”等の別は在庫情報ツリーDB上の在庫数フィールドに格納される情報に対応する。

【0024】本発明の実施の形態では、商品毎の在庫情報ツリーへの高速アクセスを可能とするために、在庫情報ツリーオンメモリ構造作成装置102によって、図3(a)及び(b)等として示される商品毎の在庫情報ツリーのオンメモリ構造が、補助記憶装置である在庫情報ツリーDB記憶装置101から半導体記憶装置である在庫情報ツリーオンメモリ構造記憶装置103に展開される。

【0025】図4及び図5は、在庫情報ツリーオンメモリ構造記憶装置103に展開される在庫情報ツリーオンメモリ構造のデータ構成図である。在庫情報ツリーオンメモリ構造においては、図4に示されるように、ノード毎に、ノードID、ノード更新フラグ、属性、値、在庫数、親ノードID、子ノードへのポインタのリストからなるデータ構造体（或いはオブジェクトインスタンス）が保持される。

【0026】また、図5に示されるように、ノードID毎に、それに対応するノードのデータ構造体のアドレスへのポインタを保持するマップテーブルが保持される。図4に示されるノード毎のデータ構造体において、ノードID、属性、値、在庫数、親ノードIDは、それぞれ、在庫情報ツリーDB記憶装置101に記憶される図2に示される在庫情報ツリーDB上のIDフィールド、属性フィールド、値フィールド、在庫数フィールド、及び親IDフィールドに対応する。

【0027】ノード更新フラグは、それが含まれるノードの在庫数が図1に示される後述する在庫引当装置104及び在庫仕入装置105により更新された場合に1にセットされる。図1に示される後述する在庫情報ツリーDB更新装置106は、在庫情報ツリーオンメモリ構造において、ノード更新フラグ=1がセットされているノードについてのみ、在庫情報ツリーDB記憶装置101に記憶されている在庫情報ツリーDBのテーブル上のそのノードに対応するレコードの在庫数フィールドを更新することにより、在庫情報ツリーDBと在庫情報ツリーオンメモリ構造との効率的な整合化を図ることができる。

【0028】子ノードへのポインタのリストは、このリストが含まれるノードを親ノードとするノードのデータ構造体のアドレスへのポインタのリストであり、在庫引

当装置 104 が、在庫情報ツリーオンメモリ構造上で、上位階層のノードから下位階層のノードへの効率的なナビゲーションを可能とするためのデータである。

【0029】また、図 5 に示されるマップテーブルは、後述する在庫情報ツリーオンメモリ構造作成装置 102 によって作成及び使用される。図 1 の在庫引当装置 104 は、在庫情報ツリーオンメモリ構造記憶装置 103 に記憶される図 4 に示されるデータ構成を有する在庫情報ツリーオンメモリ構造に対して、その在庫情報ツリーにより表現される在庫引当戦略方針に沿った自動的な在庫引当を、図 6 及び図 7 に示されるようにして実現することができる。

【0030】今、在庫引当装置 104 に対し、“サイズ L の商品 A を 250 個引き当てる”という引当条件が指定されたとする。この場合には、在庫情報ツリーオンメモリ構造記憶装置 103 上の図 6(a) に示される在庫情報ツリーオンメモリ構造がアクセスされる。

【0031】まず、在庫引当装置 104 は、在庫情報ツリーオンメモリ構造上の最上位のノードから最下位階層の末端ノードに向かって、以下に示されるナビゲーションアルゴリズムに従ってナビゲーションを実行する。

(1) 引当条件に指定されている属性に対応する下位階層へのナビゲーションの実行時には、引当条件に指定されている属性の値と同じ値が設定されているノードへのナビゲーションが実行される。

(2) 引当条件に指定されていない属性に対応する下位階層へのナビゲーションの実行時には、引当条件が満たされるまで、ツリー上の左端から右端に向かってリカーシブにナビゲーションが実行される。今、“サイズ L の商品 A を 250 個引き当てる”という引当条件が入力された場合に、図 6(a) の例では、まず、引当条件に“色”属性は指定されていないため、“商品 A”属性値が設定されている最上位のノードからそれが含まれる最上位の階層の 1 つ下の階層の左端の“色=赤”属性値が設定されている中間ノードが検索される。

【0032】続いて、更に 1 つ下の“サイズ”属性に対応する階層の各末端ノードが左端から順次検索される。その結果、引当条件に合致する“サイズ=L”属性値が設定されている末端ノードから、それに設定されている在庫数“200”に等しい数の商品 A が引き当てられる。この結果、図 6(b) に示されるように、その末端ノードの在庫数は“0”になる。

【0033】ここで、引当条件は“250”であるため、図 7(c) に示されるように、上記末端ノードからそれが含まれる階層の 1 つ上の階層の“色=赤”属性値が設定されている中間ノードに対して、不足数“50”が返され、その結果、その中間ノードにおいて、引当条件が“サイズ L の商品 A を 50 個引き当てる”に変更される。

【0034】その中間ノードの子ノードには、上記新た

な引当条件に合致する未検索の末端ノードは存在しないため、その中間ノードからそれが含まれる階層の 1 つ上の最上位の階層の“商品 A”属性値が設定されている最上位のノードに対して、上記不足数“50”が返される。

【0035】この結果、引当条件に“色”属性は指定されていないため、図 7(c) に示されるように、“商品 A”属性値が設定されている最上位のノードからそれが含まれる最上位の階層の 1 つ下の階層の未検索の“色=青”属性値が設定されている中間ノードが検索される。

【0036】続いて、更に 1 つ下の“サイズ”属性に対応する階層の各末端ノードが左端から順次検索される。その結果、引当条件に合致する“サイズ=L”属性値が設定されている末端ノードから、それに設定されている在庫数“100”に等しい数の商品 A が引き当てられる。この結果、図 7(d) に示されるように、その末端ノードの在庫数は“50”になる。

【0037】そして、図 7(c) に示されるように、上記末端ノードからそれが含まれる階層の 1 つ上の階層の“色=青”属性値が設定されている中間ノードに対して、不足数“0”が返され、更にその中間ノードからそれが含まれる階層の 1 つ上の最上位の階層の“商品 A”属性値が設定されている最上位のノードに対して、上記不足数“0”が返される。

【0038】上記最上位のノードでは、不足数“0”が判定されることにより、在庫引当が完了する。上述の在庫引当処理は、在庫情報ツリーオンメモリ構造記憶装置 103 上の在庫情報ツリーオンメモリ構造ではなく、在庫情報ツリー DB 記憶装置 101 上の在庫情報ツリー DB に対して直接行うことも可能であるが、在庫情報ツリーオンメモリ構造を介して実行される方が、高速アクセスが可能となる。

【0039】以上のようにして、在庫情報ツリーに対して、前述したナビゲーションアルゴリズムに従って在庫引当処理のナビゲーションが実行が実行されることにより、在庫引当時に条件指定が曖昧であっても、ツリー構造における検索の優先順位に対応する在庫引当戦略方針に沿った自動的な在庫引当が実現される。

在庫情報ツリーオンメモリ構造作成装置 102 の動作の詳細

図 8 は、在庫情報ツリーオンメモリ構造作成装置 102 の動作を示す動作フローチャートである。

【0040】まず、“商品”属性の特定の値（例えば商品 A）に対応する在庫情報ツリーオンメモリ構造を作成するために、在庫情報ツリー DB 記憶装置 101 に記憶される在庫情報ツリー DB 上のテーブルから、上記特定商品値に対応するレコード群を抽出するための SQL（構造化問合せ言語）文が実行される（図 8 のステップ 801）。

【0041】なお、本実施の形態では、在庫情報ツリー

DBは、SQLデータベースとして実装されている。また、本実施の形態では、特定商品値に対応するレコード群を、1つのSQL文で抽出可能とするために、在庫情報ツリーDBのテーブル構造において、図2に示される構成に加えて、図9に示されるように、“商品コード”フィールドが追加されている。

【0042】更に本実施の形態では、在庫情報ツリーの上位階層に含まれるノードほど、それに対応するレコードのIDフィールドには小さな値のノードIDが設定され、また、同一階層内では左側のノードほどそれに対応するレコードのIDフィールドには小さな値のノードIDが設定される。この結果、小さい値のノードIDを有するノードほど、在庫情報ツリー上での在庫引当処理におけるナビゲーションの優先順位が高くなる。

【0043】なお、IDフィールドの値が非連続の飛び番号に設定されることにより、ノードの追加を容易に行うことができる。図8のステップ801で実行されるSQL文は、例えば下記に示される。select from テーブル名 where 商品コード=商品A order by IDこのようにして抽出されたレコード群は、在庫情報ツリーの構造に対応してノードIDの値が小さなノードから順に並ぶ。このようなレコード群に対して、図8のステップ802～806のループ処理によって、図10(a)に示されるように先頭から順次1回だけ走査が行われることにより、図10(b)に示されるように、在庫情報ツリーの最上位の階層から下位階層に向けて順次、在庫情報ツリーオンメモリ構造を作成することができる。

【0044】即ちまず、抽出されたレコード群から、1レコードが取得される(ステップ802)。次に、レコードが取得できたか否かが判定される(ステップ803)。

【0045】レコードが取得できステップ803の判定がYESの場合には、取得されたレコードのデータを用いてノードが作成される(ステップ804)。具体的には、在庫情報ツリーオンメモリ構造記憶装置103上に、図4に示されるデータ構成を有する1つのノードのデータ構造体(或いはオブジェクトインスタンス)が確保され、そのデータ構造体のノードID、属性、値、在庫数、及び親ノードIDの各変数メンバに、在庫情報ツリーDBのテーブルから取得された上記レコード上のIDフィールド、属性フィールド、値フィールド、在庫数フィールド、及び親IDフィールドに格納されている各値がコピーされる。また、上記データ構造体のノード更新フラグに0がセットされる。

【0046】またステップ804では、作成されたノードのノードIDと、そのノードのデータ構造体へのポインタとが、図5に示されるマップテーブルに登録される。次に、図5のマップテーブルが検索されることにより、ステップ804で作成されたノードのデータ構造体に設定された親ノードIDに対応するノードのデータ構

造体が検索される(ステップ805)。

【0047】そして、ステップ805で検索されたデータ構造体の子ノードへのポインタのリストに、ステップ804で作成されたノードのデータ構造体へのポインタが登録される(ステップ806)。なお、このリストは実際には、上記データ構造体とは独立した記憶エリアに確保され、上記データ構造体にはその記憶エリアへのポインタが記憶される。

【0048】その後、ステップ802の処理に戻り、ステップ801で取得されたレコード群から次のレコードが取得され、上記ステップ802～806の一連の処理が実行される。

【0049】レコードが取得できずステップ803の判定がNOとなると、1つの商品(例えば商品A)に対応する在庫情報ツリーオンメモリ構造の作成が終了する。
在庫引当装置104の動作の詳細

図11～図14は、在庫引当装置104の動作を示す動作フローチャートであり、図11はメイン動作フローチャートである。

【0050】まず、最上位のノードに引当依頼がなされる(図11のステップ1101)。この場合の入力は引当条件と引当数であり、返却値は不足数となる。前述した図6(a)の例では、

引当条件：“サイズLの商品A”

引当数：250

である。

【0051】図12は、図11のステップ1101からコールされる引当依頼処理を示す動作フローチャートである。まず、自ノードのデータ構造体(図4)に設定されている属性が引当条件に含まれているか否かが判定される(図12のステップ1201)。図6(a)の例では、最上位の階層のノードのデータ構造体に設定されている属性“商品”は、上記引当条件に含まれているため、ステップ1201の判定はYESとなる。

【0052】ステップ1201の判定がYESの場合は、自ノードのデータ構造体に設定されている値が引当条件に合致しているか否かが判定される(図12のステップ1202)。図6(a)の例では、最上位の階層のノードのデータ構造体に設定されている値“商品A”は、上記引当条件に合致するため、ステップ1202の判定はYESとなる。

【0053】ステップ1202の判定がYESの場合には、自ノードが末端ノードであるか否かが判定される(ステップ1203)。図6(a)の例では、最上位の階層のノードは末端ノードではないため、ステップ1203の判定はNOとなる。

【0054】ステップ1203の判定がNOの場合は、中間ノードの処理が実行される(図12のステップ1204)。この場合の入力は引当条件と引当数であり、返却値は不足数となる。前述した図6(a)の例では、

引当条件: “サイズLの商品A”

引当数: 250

である。

【0055】図13は、図12のステップ1204からコールされる中間ノードの処理を示す動作フローチャートである。まず、子ノードが検索される(図13のステップ1301)。図6(a)の例では、最上位の階層のノードのデータ構造体には子ノードへのポインタのリストの先頭に、“色=赤”属性値が設定されているノードへのポインタが記憶されているため、ステップ1301の判定がYESとなる。

【0056】ステップ1301の判定がYESの場合は、対象子ノードに引当依頼がなされる(図13のステップ1302)。この場合の入力は引当条件と引当数であり、返却値は不足数となる。前述した図6(a)の例では、

引当条件: “サイズLの商品A”

引当数: 250

である。

【0057】図13のステップ1302からコールされる引当依頼処理は、前述した図12の動作フローチャートによって示される。まず、自ノードのデータ構造体に設定されている属性が引当条件に含まれているか否かが判定される(図12のステップ1201)。図6(a)の例では、“色=赤”属性値が設定されているノードのデータ構造体に設定されている属性“色”は、上記引当条件に含まれていないため、ステップ1201の判定はNOとなる。

【0058】ステップ1201の判定がNOの場合は、自ノードが末端ノードであるか否かが判定される(ステップ1203)。図6(a)の例では、“色=赤”属性値が設定されているノードは末端ノードではないため、ステップ1203の判定はNOとなる。

【0059】ステップ1203の判定がNOの場合は、中間ノードの処理が実行される(図12のステップ1204)。この場合の入力は引当条件と引当数であり、返却値は不足数となる。前述した図6(a)の例では、

引当条件: “サイズLの商品A”

引当数: 250

である。

【0060】この結果、図12のステップ1204から再び図13の動作フローチャートの中間ノードの処理がコールされる。まず、子ノードが検索される(図13のステップ1301)。図6(a)の例では、“色=赤”属性値が設定されているノードのデータ構造体には子ノードへのポインタのリストの先頭に、“サイズ=M”属性値が設定されているノードへのポインタが記憶されているため、ステップ1301の判定がYESとなる。

【0061】ステップ1301の判定がYESの場合は、前述したように、対象子ノードに引当依頼がなされ

る(図13のステップ1302)。この場合の入力は引当条件と引当数であり、返却値は不足数となる。前述した図6(a)の例では、

引当条件: “サイズLの商品A”

引当数: 250

である。

【0062】この結果、図13のステップ1302から再び図12の動作フローチャートで示される引当依頼処理がコールされる。まず、自ノードのデータ構造体に設定されている属性が引当条件に含まれているか否かが判定される(図12のステップ1201)。図6(a)の例では、“色=赤”属性値が設定されているノードの配下の“サイズ=M”属性値が設定されているノードのデータ構造体に設定されている属性“サイズ”は、上記引当条件に含まれているため、ステップ1201の判定はYESとなる。

【0063】ステップ1201の判定がYESの場合は、自ノードのデータ構造体に設定されている値が引当条件に合致しているか否かが判定される(図12のステップ1202)。図6(a)の例では、“サイズ=M”属性値は、上記引当条件に合致しないため、ステップ1202の判定はNOとなる。

【0064】ステップ1202の判定がNOの場合は、不足数に引当数がセットされて(ステップ1206)、図13のステップ1302の引当依頼処理を終了する。図6(a)の例では、不足数=250である。

【0065】図13において、ステップ1302の処理の後、不足数が引当数に等しいか否かが判定される(ステップ1303)。図6(a)の例では、“色=赤”属性値が設定されているノードの“サイズ=M”属性値が設定されている子ノードに対する引当依頼処理からのリターン時には、上述のように不足数=250=引当数であるため、ステップ1303の判定がYESとなる。

【0066】ステップ1303の判定がYESの場合には、次の子ノードが検索される(図13のステップ1301)。図6(a)の例では、“色=赤”属性値が設定されているノードのデータ構造体において子ノードへのポインタのリストの2番目に、“サイズ=L”属性値が設定されているノードへのポインタが記憶されているため、ステップ1301の判定がYESとなる。

【0067】ステップ1301の判定がYESの場合は、前述したように、対象子ノードに引当依頼がなされる(図13のステップ1302)。この場合の入力は引当条件と引当数であり、返却値は不足数となる。前述した図6(a)の例では、

引当条件: “サイズLの商品A”

引当数: 250

である。

【0068】この結果、図13のステップ1302から再び図12の動作フローチャートで示される引当依頼処

理がコールされる。まず、自ノードのデータ構造体に設定されている属性が引当条件に含まれているか否かが判定される(図12のステップ1201)。図6(a)の例では、“色=赤”属性値が設定されているノードの配下の“サイズ=L”属性値が設定されているノードのデータ構造体に設定されている属性“サイズ”は、上記引当条件に含まれているため、ステップ1201の判定はYESとなる。

【0069】ステップ1201の判定がYESの場合は、自ノードのデータ構造体に設定されている値が引当条件に合致しているか否かが判定される(図12のステップ1202)。図6(a)の例では、“サイズ=L”属性値は、上記引当条件に合致するため、ステップ1202の判定はYESとなる。

【0070】ステップ1202の判定がYESの場合には、自ノードが末端ノードであるか否かが判定される(ステップ1203)。図6(a)の例では、“色=赤”属性値が設定されているノードの配下の“サイズ=L”属性値が設定されているノードは末端ノードであるため、ステップ1203の判定はNOとなる。

【0071】ステップ1203の判定がYESの場合には、末端ノードの処理が実行される(図12のステップ1205)。この場合の入力は引当条件と引当数であり、返却値は不足数となる。前述した図6(a)の例では、

引当条件：“サイズLの商品A”

引当数：250

である。

【0072】図14は、図12のステップ1205からコールされる末端ノードの処理を示す動作フローチャートである。まず、その末端ノードのデータ構造体に設定されている在庫数が引当数より多いか否かが判定される(図14のステップ1401)。図6(a)の例では、“色=赤”属性値が設定されているノードの配下の“サイズ=L”属性値が設定されている末端ノードのデータ構造体に設定されている在庫数=200は引当数=250より少ないため、ステップ1401の判定はNOとなる。

【0073】ステップ1401の判定がNOの場合には、引当数から現在の在庫数を減算して得られる値が不足数にセットされ、新たな在庫数が0にされる(図14のステップ1403)。図6(a)→図6(b)→図7(c)の例では、“色=赤”属性値が設定されているノードの配下の“サイズ=L”属性値が設定されている末端ノードの処理においては、

不足数=引当数-在庫数=250-200=50

とされ、データ構造体には新たな在庫数として、

在庫数=0

がセットされる。

【0074】その後、上記末端ノードのデータ構造体の

ノード更新フラグが1にセットされて(図14のステップ1404)、図12のステップ1205の末端ノードの処理を終了し、続いて、図13のステップ1302の引当依頼処理を終了する。

【0075】図13において、ステップ1302の処理の後、不足数が引当数に等しいか否かが判定される(ステップ1303)。図6(a)の例では、色=赤”属性値が設定されているノードの“サイズ=L”属性値が設定されている子ノードに対する引当依頼処理からのリターン時には、上述のように不足数=50≠引当数=250であるため、ステップ1303の判定がNOとなる。

【0076】ステップ1303の判定がNOの場合には、現在の中間ノードのデータ構造体のノード更新フラグに1がセットされる(図13のステップ1304)。即ち、配下のノードで在庫数が更新され、その親ノードである自ノードの在庫数も次のように更新されるため、ノード更新フラグがセットされることになる。図7(c)の例では、色=赤”属性値が設定されているノードのデータ構造体のノード更新フラグが1にセットされる。

【0077】続いて、引当数から不足数を減算して得られる値を、現在の中間ノードのデータ構造体の在庫数から減算して得られる値が、新たな在庫数としてセットされ、また、不足数が新たな引当数にセットされる(図13のステップ1305)。図6(b)→図7(c)の例では、色=赤”属性値が設定されているノードのデータ構造体の在庫数について、

在庫数=在庫数-(引当数-不足数)

=600-(250-50)=400

となり、新たな引当数は、

引当数=不足数=50

となる。

【0078】上記処理の後、不足数が0になったか否かが判定される(図13のステップ1306)。上記例では、不足数=50≠0であるため、ステップ1306の判定はNOとなる。

【0079】ステップ1306の判定がNOの場合は、次の子ノードが検索される(図13のステップ1301)。図6(a)の例では、“色=赤”属性値が設定されているノードのデータ構造体において子ノードへのポインタのリストの3番目に、“サイズ=S”属性値が設定されているノードへのポインタが記憶されているため、ステップ1301の判定がYESとなる。

【0080】ステップ1301の判定がYESの場合は、前述したように、対象子ノードに引当依頼がなされる(図13のステップ1302)。この場合の入力は引当条件と引当数であり、返却値は不足数となる。図7(c)の例では、

引当条件：“サイズLの商品A”

引当数：50

となる。

【0081】この結果、図13のステップ1302から再び図12の動作フローチャートで示される引当依頼処理がコールされる。まず、自ノードのデータ構造体に設定されている属性が引当条件に含まれているか否かが判定される(図12のステップ1201)。図7(c)の例では、“色=赤”属性値が設定されているノードの配下の“サイズ=S”属性値が設定されているノードのデータ構造体に設定されている属性“サイズ”は、上記引当条件に含まれているため、ステップ1201の判定はYESとなる。

【0082】ステップ1201の判定がYESの場合は、自ノードのデータ構造体に設定されている値が引当条件に合致しているか否かが判定される(図12のステップ1202)。図7(c)の例では、“サイズ=S”属性値は、上記引当条件に合致しないため、ステップ1202の判定はNOとなる。

【0083】ステップ1202の判定がNOの場合は、不足数に引当数がセットされて(ステップ1206)、図13のステップ1302の引当依頼処理を終了する。図7(c)の例では、不足数=50である。

【0084】図13において、ステップ1302の処理の後、不足数が引当数に等しいか否かが判定される(ステップ1303)。図7(c)の例では、色=赤”属性値が設定されているノードの“サイズ=S”属性値が設定されている子ノードに対する引当依頼処理からのリターン時には、上述のように不足数=50=引当数であるため、ステップ1303の判定がYESとなる。

【0085】ステップ1303の判定がYESの場合は、次の子ノードが検索される(図13のステップ1301)。図7(c)の例では、“色=赤”属性値が設定されているノードのデータ構造体において子ノードへのポインタのリストの3番目以降にはポインタは記憶されていないため、ステップ1301の判定がNOとなる。

【0086】ステップ1301の判定がNOの場合には、図12のステップ1204の中間ノードの処理を終了し、更に図13のステップ1302の引当依頼処理を終了する。図7(c)の例では、“色=赤”属性値が設定されている中間ノードに対する引当依頼処理が終了し、制御が、最上の階層のノードについての図12の動作フローチャートで示される中間ノードの処理のステップ1302の直後に戻る。

【0087】図13において、ステップ1302の処理の後、不足数が引当数に等しいか否かが判定される(ステップ1303)。図7(c)の例では、最上位の階層のノードの“色=赤”属性値が設定されている子ノードに対する引当依頼処理からのリターン時には、不足数=50であり、また、引当数は、図6(a)の状態であるから250である。このため、ステップ1303の判定がNOとなる。

【0088】ステップ1303の判定がNOの場合に

は、前述したように、現在の中間ノードのデータ構造体のノード更新フラグに1がセットされる(図13のステップ1304)。図7(c)の例では、最上位の階層のノードのデータ構造体のノード更新フラグが1にセットされる。

【0089】続いて、引当数から不足数を減算して得られる値を、現在の中間ノードのデータ構造体の在庫数から減算して得られる値が、新たな在庫数としてセットされ、また、不足数が新たな引当数にセットされる(図13のステップ1305)。図6(b)→図7(c)の例では、最上位の階層のノードのデータ構造体の在庫数について、

$$\begin{aligned} \text{在庫数} &= \text{在庫数} - (\text{引当数} - \text{不足数}) \\ &= 800 - (250 - 50) = 600 \end{aligned}$$

となり、新たな引当数は、

$$\text{引当数} = \text{不足数} = 50$$

となる。

【0090】上記処理の後、不足数が0になったか否かが判定される(図13のステップ1306)。上記例では、不足数=50≠0であるため、ステップ1306の判定はNOとなる。

【0091】ステップ1306の判定がNOの場合は、次の子ノードが検索される(図13のステップ1301)。図7(c)の例では、最上位の階層のノードのデータ構造体において子ノードへのポインタのリストの2番目に、“色=青”属性値が設定されているノードへのポインタが記憶されているため、ステップ1301の判定がYESとなる。

【0092】ステップ1301の判定がYESの場合は、前述したように、対象子ノードに引当依頼がなされる(図13のステップ1302)。この場合の入力は引当条件と引当数であり、返却値は不足数となる。図7(c)の例では、

引当条件：“サイズLの商品A”

引当数：50

となる。

【0093】この結果、図13のステップ1302から再び図12の動作フローチャートで示される引当依頼処理がコールされる。まず、自ノードのデータ構造体に設定されている属性が引当条件に含まれているか否かが判定される(図12のステップ1201)。図7(c)の例では、“色=青”属性値が設定されているノードのデータ構造体に設定されている属性“色”は、上記引当条件に含まれていないため、ステップ1201の判定はNOとなる。

【0094】ステップ1201の判定がNOの場合は、自ノードが末端ノードであるか否かが判定される(ステップ1203)。図7(c)の例では、“色=青”属性値が設定されているノードは末端ノードではないため、ステップ1203の判定はNOとなる。

【0095】ステップ1203の判定がNOの場合は、中間ノードの処理が実行される（図12のステップ1204）。この場合の入力は引当条件と引当数であり、返却値は不足数となる。前述した図6(a)の例では、引当条件：“サイズLの商品A”

引当数：50

である。

【0096】この結果、図12のステップ1204から再び図13の動作フローチャートの中間ノードの処理がコールされる。まず、子ノードが検索される（図13のステップ1301）。図7(c)の例では、“色=青”属性値が設定されているノードのデータ構造体には子ノードへのポインタのリストの先頭に、“サイズ=L”属性値が設定されているノードへのポインタが記憶されているため、ステップ1301の判定がYESとなる。

【0097】ステップ1301の判定がYESの場合は、前述したように、対象子ノードに引当依頼がなされる（図13のステップ1302）。この場合の入力は引当条件と引当数であり、返却値は不足数となる。前述した図6(a)の例では、

引当条件：“サイズLの商品A”

引当数：50

である。

【0098】この結果、図13のステップ1302から再び図12の動作フローチャートで示される引当依頼処理がコールされる。まず、自ノードのデータ構造体に設定されている属性が引当条件に含まれているか否かが判定される（図12のステップ1201）。図7(c)の例では、“色=青”属性値が設定されているノードの配下の“サイズ=L”属性値が設定されているノードのデータ構造体に設定されている属性“サイズ”は、上記引当条件に含まれているため、ステップ1201の判定はYESとなる。

【0099】ステップ1201の判定がYESの場合は、自ノードのデータ構造体に設定されている値が引当条件に合致しているか否かが判定される（図12のステップ1202）。図7(c)の例では、“サイズ=L”属性値は、上記引当条件に合致するため、ステップ1202の判定はYESとなる。

【0100】ステップ1202の判定がYESの場合には、自ノードが末端ノードであるか否かが判定される（ステップ1203）。図7(c)の例では、“色=青”属性値が設定されているノードの配下の“サイズ=L”属性値が設定されているノードは末端ノードであるため、ステップ1203の判定はNOとなる。

【0101】ステップ1203の判定がYESの場合には、末端ノードの処理が実行される（図12のステップ1205）。この場合の入力は引当条件と引当数であり、返却値は不足数となる。前述した図7(c)の例では、

引当条件：“サイズLの商品A”

引当数：50

である。

【0102】末端ノードの処理は、前述した図14の動作フローチャートで示される。まず、その末端ノードのデータ構造体に設定されている在庫数が引当数より多いか否かが判定される（図14のステップ1401）。図7(c)の例では、“色=青”属性値が設定されているノードの配下の“サイズ=L”属性値が設定されている末端ノードのデータ構造体に設定されている在庫数=100は引当数=50より多いため、ステップ1401の判定はYESとなる。

【0103】ステップ1401の判定がYESの場合には、在庫数から引当数を減算して得られる値が新たな在庫数としてセットされると共に、不足数が0にされる（図14のステップ1402）。図7(c)→図7(d)の例では、“色=青”属性値が設定されているノードの配下の“サイズ=L”属性値が設定されている末端ノードの処理においては、

20 在庫数=在庫数-引当数=100-50=50

不足数=0

とされる。

【0104】その後、上記末端ノードのデータ構造体のノード更新フラグが1にセットされて（図14のステップ1404）、図12のステップ1205の末端ノードの処理を終了し、続いて、図13のステップ1302の引当依頼処理を終了する。

【0105】図13において、ステップ1302の処理の後、不足数が引当数に等しいか否かが判定される（ステップ1303）。図7(d)の例では、色=青”属性値が設定されているノードの“サイズ=L”属性値が設定されている子ノードに対する引当依頼処理からのリターン時には、上述のように不足数=0≠引当数=50であるため、ステップ1303の判定がNOとなる。

【0106】ステップ1303の判定がNOの場合は、現在の間中ノードのデータ構造体のノード更新フラグに1がセットされる（図13のステップ1304）。図7(d)の例では、色=青”属性値が設定されているノードのデータ構造体のノード更新フラグが1にセットされる。

【0107】続いて、引当数から不足数を減算して得られる値を、現在の間中ノードのデータ構造体の在庫数から減算して得られる値が、新たな在庫数としてセットされ、また、不足数が新たな引当数にセットされる（図13のステップ1305）。図7(c)→図7(d)の例では、色=青”属性値が設定されているノードのデータ構造体の在庫数について、

在庫数=在庫数-(引当数-不足数)

=200-(50-0)=150

50 となり、新たな引当数は、

引当数=不足数=0

となる。

【0108】上記処理の後、不足数が0になったか否かが判定される(図13のステップ1306)。上記例では、不足数=0であるため、ステップ1306の判定はYESとなる。

【0109】ステップ1306の判定がYESの場合は、図12のステップ1204の中間ノードの処理を終了し、更に図13のステップ1302の引当依頼処理を終了する。図7(d)の例では、“色=青”属性値が設定されている中間ノードに対する引当依頼処理が終了し、制御が、最上の階層のノードについての図12の動作フローチャートで示される中間ノードの処理のステップ1302の直後に戻る。

【0110】図13において、ステップ1302の処理の後、不足数が引当数に等しいか否かが判定される(ステップ1303)。図7(d)の例では、最上位の階層のノードの“色=青”属性値が設定されている子ノードに対する引当依頼処理からのリターン時には、不足数=0であり、また、引当数は、図7(c)の状態であるから50である。このため、ステップ1303の判定がYESとなる。

【0111】ステップ1303の判定がNOの場合には、前述したように、現在の中間ノードのデータ構造体のノード更新フラグに1がセットされる(図13のステップ1304)。図7(d)の例では、最上位の階層のノードのデータ構造体のノード更新フラグが1に再度セットされる。

【0112】続いて、引当数から不足数を減算して得られる値を、現在の中間ノードのデータ構造体の在庫数から減算して得られる値が、新たな在庫数としてセットされ、また、不足数が新たな引当数にセットされる(図13のステップ1305)。図7(c)→図7(d)の例では、最上位の階層のノードのデータ構造体の在庫数について、

在庫数=在庫数-(引当数-不足数)

=600-(50-0)=550

となり、新たな引当数は、

引当数=不足数=0

となる。

【0113】上記処理の後、不足数が0になったか否かが判定される(図13のステップ1306)。上記例では、不足数=0であるため、ステップ1306の判定はYESとなる。

【0114】ステップ1306の判定がYESの場合には、図12のステップ1204の中間ノードの処理を終了し、更に図13のステップ1302の引当依頼処理を終了する。図7(d)の例では、最上位のノードに対する引当依頼処理が終了し、制御が、図11の動作フローチャートで示される引当処理のステップ1101の直後に

戻る。

【0115】図11において、不足数が0であるか否かが判定される(図11のステップ1102)。図7(d)の例では、不足数=0であるため、ステップ1102の判定はYESとなる。

【0116】ステップ1102の判定がYESの場合には、引当成功が通知されて、在庫引当装置104による在庫引当処理が終了する。一方、ステップ1102の判定がNOの場合には、ノード更新が全て破棄され(図11のステップ1103)、在庫情報ツリーオンメモリ構造記憶装置103の記憶内容は変化しない。

【0117】そして、引当不能が通知されて、在庫引当装置104による在庫引当処理が終了する。なお、中間ノードのデータ構造体に設定されている属性と値が引当条件に合致している場合であって(図12のステップ1201の判定がYES→1202の判定がYES→1203の判定がNO→1204)、その中間ノードの処理(図13)における対象子ノードに対する引当依頼処理(図13のステップ1302、図12)において、その子ノードが末端ノードである場合には、その末端ノードのデータ構造体に設定されている属性が引当条件に含まれていない場合であっても、末端ノードの処理(図12のステップ1205、図14)において、在庫引当処理が実行される。

分散化が可能な在庫引当処理

上述の在庫引当処理においては、末端ノードにおいて、在庫数が0になるまでの限度で在庫引当処理が実行される。

【0118】これに対して、中間ノードのデータ構造体に設定されている属性と値が引当条件に合致している場合であって、その中間ノードの処理における対象子ノードである末端ノードに対する引当依頼処理において、その末端ノードのデータ構造体に設定されている属性が引当条件に含まれていない場合には、所定の在庫数下限までの限度で在庫引当処理が実行されるように構成することもできる。

【0119】図15は、図12を置き換える引当依頼処理の動作フローチャートである。図15において、図12の場合と同じ番号が付されたステップは、図12の場合と同じ機能を実現する。

【0120】図12の動作フローチャートと異なる点として、図15の動作フローチャートでは、中間ノードのデータ構造体に設定されている属性と値が引当条件に合致している場合であって(図15のステップ1501の判定がYES→1502の判定がYES→1503の判定がNO→1504)、その中間ノードの処理(図13)における対象子ノードに対する引当依頼処理(図13のステップ1302、図15)において、子ノードのデータ構造体に設定されている属性が引当条件に含まれておらず(図15のステップ1201の判定がNO)、

かつ自ノードが末端ノードである場合（図15のステップ1501）には、ステップ1205とは異なる在庫数下限までの限度での末端ノードの処理が実行される（図15のステップ1502）。

【0121】図16は、図15のステップ1502の末端ノードの処理を示す動作フローチャートである。まず、その末端ノードのデータ構造体に設定されている在庫数から所定の在庫数下限を減算して得られる数が引当数より多いか否かが判定される（図16のステップ1601）。

【0122】ステップ1601の判定がYESの場合には、図14のステップ1402の場合と同様に、在庫数から引当数を減算して得られる値が新たな在庫数としてセットされると共に、不足数が0にされる（図16のステップ1602）。

【0123】一方、ステップ1601の判定がNOの場合には、在庫数から在庫数下限を減算して得られる値を、引当数から減算して得られる値が不足数にセットされ、新たな在庫数が在庫数下限にされる（図16のステップ1603）。

【0124】ステップ1604の処理は図14のステップ1404の処理と同様である。以上示したように、在庫数下限までの限度での末端ノードの処理が実行されることにより、特定のノードに偏って在庫引当が行われてしまうという傾向を回避することができ、在庫引当の分散化を図ることができる。在庫仕入装置105の動作の詳細

在庫仕入が発生した場合には、在庫仕入装置105が在庫情報ツリーオンメモリ構造記憶装置103に記憶されている在庫情報ツリーオンメモリ構造に対して在庫仕入処理を実行する。

【0125】在庫仕入処理においては、仕入条件に合致するノードは、ただ1つの末端ノードと、その末端ノードが含まれる上位の中間ノードである。図17～図19は、在庫仕入装置105の動作を示す動作フローチャートであり、図17はメイン動作フローチャートである。

【0126】まず、最上位のノードに仕入依頼がなされる（図17のステップ1701）。この場合の入力は仕入条件と仕入数であり、返却値はOK又はNGとなる。図18は、図17のステップ1701からコールされる仕入依頼処理を示す動作フローチャートである。

【0127】仕入依頼処理においては、現在処理中のノードのデータ構造体に設定されている属性と値が仕入条件に合っていないければ、即座にNGが返答される（図18のステップ1801の判定がNO→1811、又はステップ1801の判定がYES→1802の判定がNO→1811）。

【0128】逆に、現在処理中のノードのデータ構造体に設定されている属性と値が仕入条件に合っており、かつ自ノードが中間ノードである場合には、中間ノードの

処理が実行される（図18のステップ1801の判定がYES→1802の判定がYES→1803の判定がNO→1804）。この場合の入力は仕入条件と仕入数であり、返却値はOK又はNGである。

【0129】図19は、図18のステップ1804からコールされる中間ノードの処理を示す動作フローチャートである。中間ノードの処理においては、そのデータ構造体の子ノードへのポインタのリストが検索されながら、対象子ノードに対して仕入依頼処理が実行される

10 （図19のステップ1901の判定がYES→1902→1903→1901のループ処理）。

【0130】対象子ノードに対する仕入依頼処理では、図18の動作フローチャートが再度実行され、子ノードが中間ノードであれば再帰的に中間ノードの処理が実行される（図18のステップ1801の判定がYES→1802の判定がYES→1803の判定がNO→1804）。

20 【0131】一方、子ノードが末端ノードであって、かつそのデータ構造体に設定されている属性と値が仕入条件に合っていれば（図18のステップ1801の判定がYES→1802の判定がYES→1803の判定がYES）、その末端ノードのデータ構造体の在庫数に仕入数が加算され（図18のステップ1808）、そのデータ構造体のノード更新フラグに1がセットされ（図18のステップ1809）、親ノードに対してOKが返答される（図18のステップ1810）。

30 【0132】末端ノードからOKが返答されると、図19の中間ノードの処理において、ステップ1903の判定がOKとなって、その中間ノードのデータ構造体のノード更新フラグに1がセットされ（図19のステップ1904）、そのデータ構造体の在庫数に仕入数が加算され（図19のステップ1905）、OKが返答される（図19のステップ1906）。

【0133】一方、中間ノードの処理において、何れの子ノードからもOKが返答されず、ステップ1901の判定がNOとなった場合には、NGが返答される（図19のステップ1907）。

40 【0134】中間ノードの処理においてOKが返答された場合には、その中間ノードに対して仕入依頼処理を実行させた親ノードに対してOKが返答され（図18のステップ1804→1805の判定がOK→ステップ1806）、逆に、NGが返答された場合には、その中間ノードに対して仕入依頼処理を実行させた親ノードに対してNGが返答される（図18のステップ1804→1805の判定がNG→ステップ1807）。

50 【0135】以上のようにして、仕入処理の結果が在庫情報ツリーオンメモリ構造の最上位の階層のノードまで戻ってくると、図17のステップ1701の仕入依頼処理が終了し、OKが返されたか否かが判定される（図17のステップ1702）。

【0136】ステップ1702でOKが返されたと判定された場合には、仕入成功が通知されて、在庫仕入装置105による在庫仕入処理が終了する。一方、ステップ1702でNGが返されたと判定された場合には、ノード更新が全て破棄され（図17のステップ1703）、在庫情報ツリーオンメモリ構造記憶装置103の記憶内容は変化しない。

【0137】そして、引当不能が通知されて、在庫仕入装置105による在庫仕入処理が終了する。

在庫情報ツリーDB更新装置106の動作の詳細

在庫情報ツリーオンメモリ構造記憶装置103に記憶される在庫情報ツリーオンメモリ構造の更新内容は、一定期間毎に在庫情報ツリーDB更新装置106によって在庫情報ツリーDB記憶装置101に記憶される在庫情報ツリーDBに反映させられる。

【0138】前述したように、在庫情報ツリーオンメモリ構造における各ノードに対応するデータ構造体に記憶されるノード更新フラグは、それが含まれるノードの在庫数が在庫引当装置104及び在庫仕入装置105により更新された場合に1にセットされる。在庫情報ツリーDB更新装置106は、在庫情報ツリーオンメモリ構造において、図24に示されるように、ノード更新フラグ=1がセットされているノードについてのみ、在庫情報ツリーDB記憶装置101に記憶されている在庫情報ツリーDBのテーブルに対して、データ更新のためのSQL文を実行する。これにより、在庫情報ツリーDBと在庫情報ツリーオンメモリ構造との効率的な整合化を図ることができる。

【0139】図20～図23は、在庫情報ツリーDB更新装置106の動作を示す動作フローチャートであり、図20はメイン動作フローチャートである。まず最上位のノードに更新依頼がなされる（図20のステップ2001）。

【0140】図21は、図20のステップ2001からコールされる更新依頼処理を示す動作フローチャートである。更新依頼処理においては、自ノードが中間ノードであれば中間ノードの更新処理が実行され（図21のステップ2101の判定がNO→2102）、自ノードが末端ノードであれば末端ノードの更新処理が実行される（ステップ2101の判定がYES→2103）。

【0141】図22は、図21のステップ2102からコールされる中間ノードの更新処理を示す動作フローチャートである。中間ノードの更新処理においては、まず、自ノードのデータ構造体に記憶されているノード更新フラグの値が1なら、そのノードに関する以下に示されるSQL文が発行される（図22のステップ2201の判定がYES→2202）。update テーブル名 set 在庫数=<在庫数> where ID=<ノードID>上記SQL文において、<在庫数>と<ノードID>は、それぞれデータ構造体に記憶されている在庫数及びノード

IDである。

【0142】その後、自ノードのデータ構造体に記憶されているノード更新フラグの値が0に戻される（ステップ2203）。続いて、自ノードのデータ構造体の子ノードへのポインタのリストが検索されながら、対象子ノードに対して更新依頼処理が実行される（図22のステップ2204の判定がYES→2205→2204のループ処理）。

【0143】対象子ノードに対する更新依頼処理では、図21の動作フローチャートが再度実行され、子ノードが中間ノードであれば再帰的に中間ノードの更新処理が実行される（図21のステップ2101の判定がNO→2202）。

【0144】一方、更新依頼処理において、自ノードが末端ノードであれば末端ノードの更新処理が実行される（ステップ2101の判定がYES→2103）。図23は、図21のステップ2103からコールされる末端ノードの更新処理を示す動作フローチャートである。

【0145】末端ノードの更新処理では、自ノードのデータ構造体に記憶されているノード更新フラグの値が1でなければ何も実行されず、ノード更新フラグの値が1であるなら、そのノードに関して中間ノードの場合と同様の以下に示されるSQL文が発行される（図23のステップ2301の判定がYES→2302）。

update テーブル名 set 在庫数=<在庫数> where ID=<ノードID>

その後、自ノードのデータ構造体に記憶されているノード更新フラグの値が0に戻される（ステップ2303）。

【0146】以上のようにして、更新処理の再帰処理が終了して最上位の階層のノードに対する更新依頼処理に制御が戻ってくると、図20のステップ2001の更新処理が終了し、在庫情報ツリーDB更新装置106による更新処理が終了する。

在庫情報ツリーDBの他の実施の形態

図25(a)は、在庫情報ツリーDB記憶装置101に記憶される在庫情報ツリーDBの他の実施の形態を示す図である。

【0147】図2又は図9に示される在庫情報ツリーDBの実施の形態では、在庫情報ツリーの上位階層に含まれるノードほど、それに対応するレコードのIDフィールドには小さな値のノードIDが設定され、また、同一階層内では左側のノードほどそれに対応するレコードのIDフィールドには小さな値のノードIDが設定されることにより、小さい値のノードIDを有するノードほど、在庫情報ツリー上での在庫引当処理におけるナビゲーションの優先順位が高くなるように構成される。

【0148】これに対して、図25(a)に示される他の実施の形態では、異階層間の優先順位は、親IDフィールドでの設定関係によって規定され、また、同一階層内

での優先順位は、新たに設けられる前ノードIDでの設定関係によって規定されるように、在庫情報ツリーDBが構成される。

【0149】この結果、前ノードIDフィールドでの設定内容に従って、例えば図25(b)に示されるようなナビゲーションの優先順位を規定することができる。また、このような構成により、図26に示されるように、任意の階層に対して任意の優先順位のノードを簡単に追加等することが可能となる。

本実施の形態を実現するプログラムが記録された記録媒体についての補足

本発明は、コンピュータにより使用されたときに、上述の本発明の実施の形態の各構成によって実現される機能と同様の機能をコンピュータに行わせるためのコンピュータ読出し可能記録媒体として構成することもできる。

【0150】この場合に、図27に示されるように、例えばフロッピーディスク、CD-ROMディスク、光ディスク、リムーバブルハードディスク等の可搬型記録媒体2402や、ネットワーク回線2703経由で、本発明の実施の形態の各種機能を実現するプログラムが、NMSサーバ2701の本体2704内のメモリ(RAM又はハードディスク等)2705にロードされて、実行される。

【0151】

【発明の効果】本発明によれば、在庫引当時に条件指定が曖昧であっても、2進木データの木構造における検索の優先順位に対応する在庫引当戦略方針に沿った自動的な在庫引当を実現することが可能となる。

【0152】また、データベーステーブルからメモリ装置上に2進木データを展開し、その2進木データに対して在庫引当又は在庫仕入を実行することにより、2進木データへの高速アクセスが可能となる。

【0153】また、データベース上の各レコード上の識別情報フィールドの内容に従ってデータベーステーブルからメモリ装置上に2進木データを展開することにより、データベーステーブルに対する簡単な問合せ文によって、データベーステーブルからメモリ装置上に2進木データを展開することが可能となる。

【0154】また、データベース上の各レコード上の管理対象種別フィールドの内容に従って、データベーステーブルからメモリ装置上に管理対象種別毎の2進木データを展開することにより、やはり、データベーステーブルに対する簡単な問合せ文によって、データベーステーブルからメモリ装置上に2進木データを展開することが可能となる。

【0155】更に、2進木データを構成する各ノードに、その各ノードに対応する在庫数が更新されたか否かを示すノード更新情報を保持し、そのノード更新情報に従ってメモリ装置上に展開された2進木データの更新状態がデータベーステーブルに反映させられることによ

り、メモリ装置上の2進木データとデータベーステーブルとの効率的な整合化を図ることが可能となる。

【図面の簡単な説明】

【図1】本発明の実施の形態のシステム構成図である。

【図2】在庫情報ツリーDBのデータ構成例(その1)を示す図である。

【図3】在庫情報ツリーの例を示す図である。

【図4】在庫情報ツリーオンメモリ構造のデータ構成図(その1)である。

10 【図5】在庫情報ツリーオンメモリ構造のデータ構成図(その2)である。

【図6】在庫引当の手順例を示す図(その1)である。

【図7】在庫引当の手順例を示す図(その2)である。

【図8】在庫情報ツリーオンメモリ構造作成装置の動作フローチャートである。

【図9】在庫情報ツリーDBのデータ構成例(その2)を示す図である。

【図10】在庫情報ツリーオンメモリ構造作成処理の説明図である。

20 【図11】引当処理のメイン動作フローチャートである。

【図12】引当依頼処理の動作フローチャートである。

【図13】引当処理における中間ノードの処理の動作フローチャートである。

【図14】引当処理における末端ノードの処理の動作フローチャートである。

【図15】分散化が可能な引当依頼処理の動作フローチャートである。

30 【図16】引当処理における在庫数下限がある末端ノードの処理の動作フローチャートである。

【図17】仕入処理のメイン動作フローチャートである。

【図18】仕入依頼処理の動作フローチャートである。

【図19】仕入処理における中間ノードの処理の動作フローチャートである。

【図20】更新処理のメイン動作フローチャートである。

【図21】更新依頼処理の動作フローチャートである。

40 【図22】更新処理における中間ノードの処理の動作フローチャートである。

【図23】更新処理における末端ノードの処理の動作フローチャートである。

【図24】更新処理の動作説明図である。

【図25】在庫情報ツリーDBの他の実施の形態の構成図である。

【図26】在庫情報ツリーDBの他の実施の形態の説明図である。

【図27】本実施の形態を実現するプログラムが記録された記録媒体の説明図である。

50 【符号の説明】

27

28

- 101 在庫情報ツリーDB記憶装置
 102 在庫情報ツリーオンメモリ構造作成装置
 103 在庫情報ツリーオンメモリ構造記憶装置

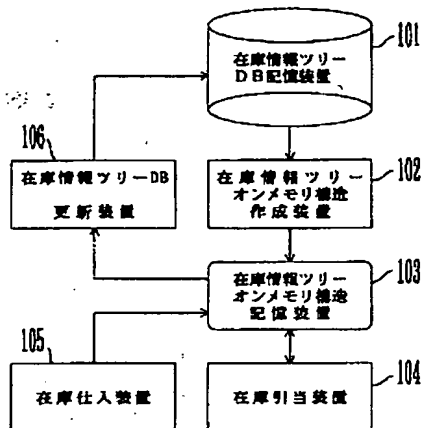
- 104 在庫引当装置
 105 在庫仕入装置
 106 在庫情報ツリーDB更新装置

【図1】

【図2】

【図5】

本発明の実施の形態のシステム構成図 在庫情報ツリーDBのデータ構成例(その1)を示す図 在庫情報ツリーDBの構造の例(その2)



ID	属性	値	在庫数	親ID
1	商品	商品 A	800	0
2	商品	商品 B	1200	0
3	色	赤	600	1
4	色	青	200	1
5	サイズ	M	300	3
6	サイズ	L	200	3
7	サイズ	S	100	3
8	サイズ	L	100	4
9	サイズ	M	50	4
10	サイズ	S	50	4
11	サイズ	M	1000	2
12	サイズ	S	200	2
13	サイズ	L	200	2
14	色	赤	400	11
15	色	青	400	11
16	色	黄	200	11

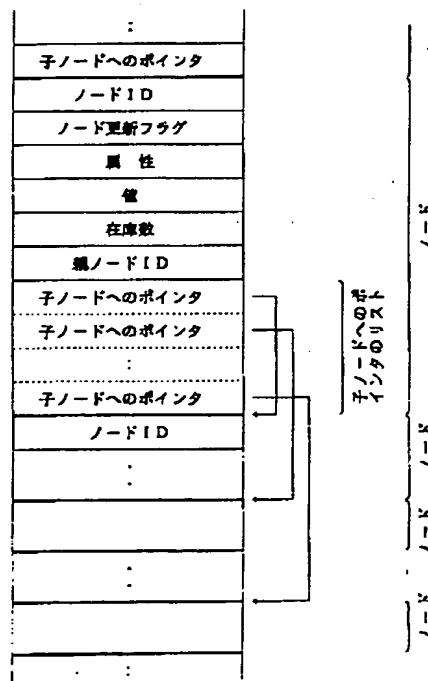
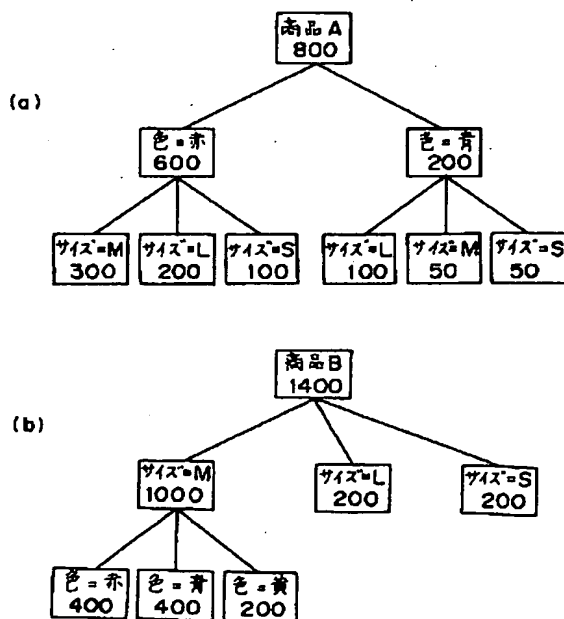
...	...
ノードID	ノードへのポインタ
ノードID	ノードへのポインタ
...	...

【図4】

【図3】

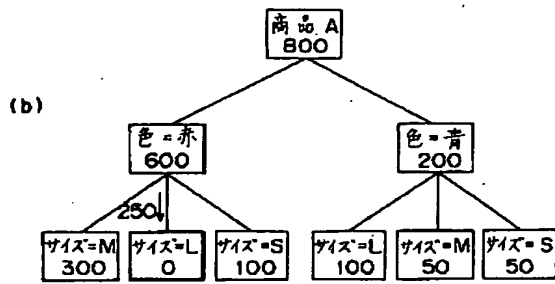
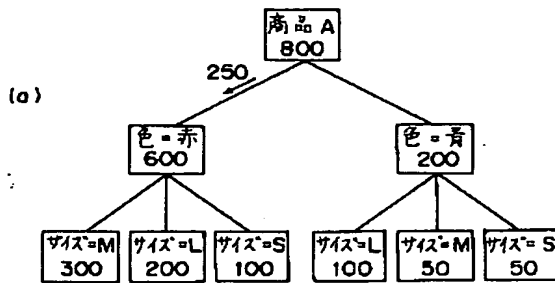
在庫情報ツリーDBの構造の例(その1)

在庫情報ツリーの例を示す図



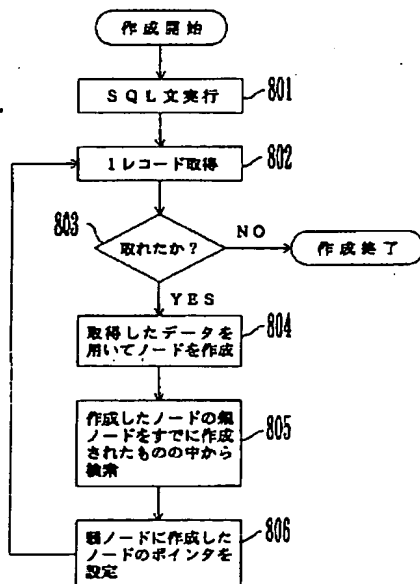
【図6】

在庫引当の手順例を示す図 (その1)



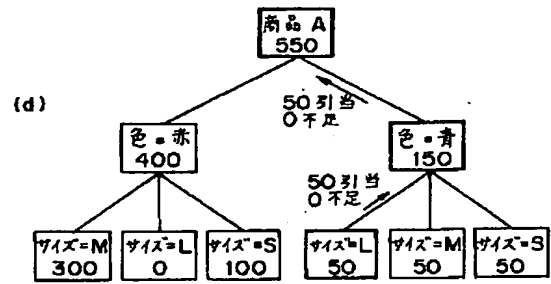
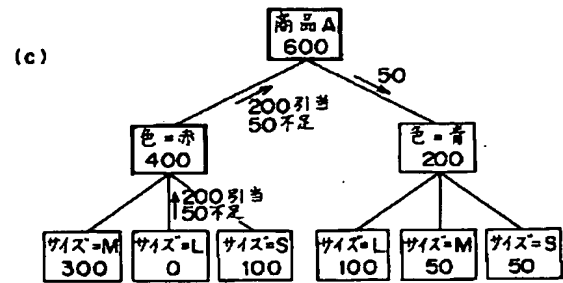
【図8】

在庫情報ツリーDB構築作成装置の動作フローチャート



【図7】

在庫引当の手順例を示す図 (その2)



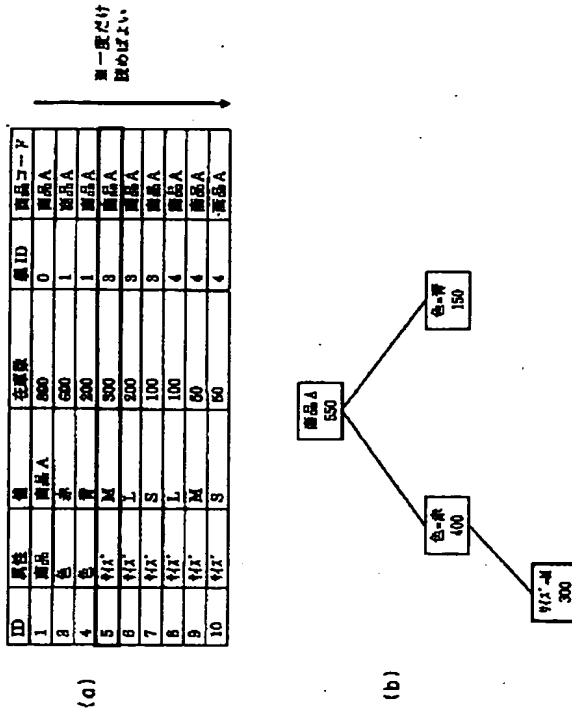
【図9】

在庫情報ツリーDBのデータ構成例(その2)を示す図

ID	属性	値	在庫数	親ID	商品コード
1	商品	商品 A	800	0	商品 A
2	商品	商品 B	1200	0	商品 B
3	色	赤	600	1	商品 A
4	色	青	200	1	商品 A
5	サイズ	M	300	3	商品 A
6	サイズ	L	200	3	商品 A
7	サイズ	S	100	3	商品 A
8	サイズ	L	100	4	商品 A
9	サイズ	M	50	4	商品 A
10	サイズ	S	50	4	商品 A
11	サイズ	M	1000	2	商品 B
12	サイズ	S	200	2	商品 B
13	サイズ	L	200	2	商品 B
14	色	赤	400	11	商品 B
15	色	青	400	11	商品 B
16	色	青	200	11	商品 B

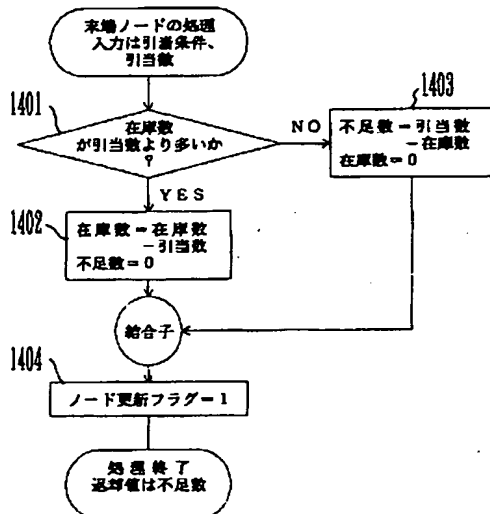
【図10】

在庫情報ツリーオンメモリ構造作成処理の説明図



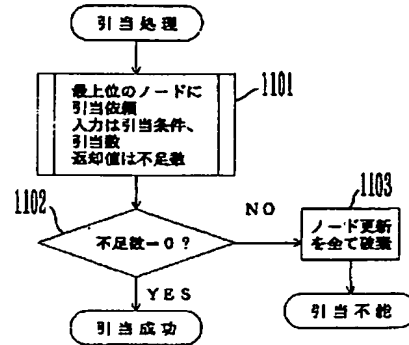
【図14】

末端ノードの処理の動作フローチャート



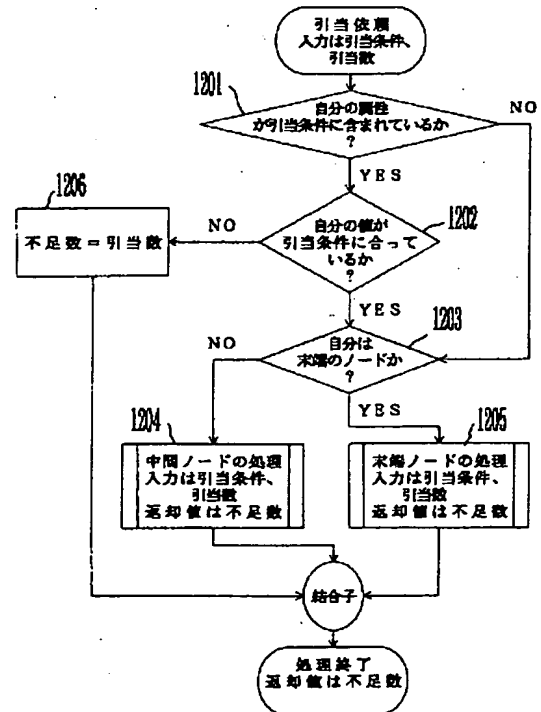
【図11】

引当処理のメイン動作フローチャート



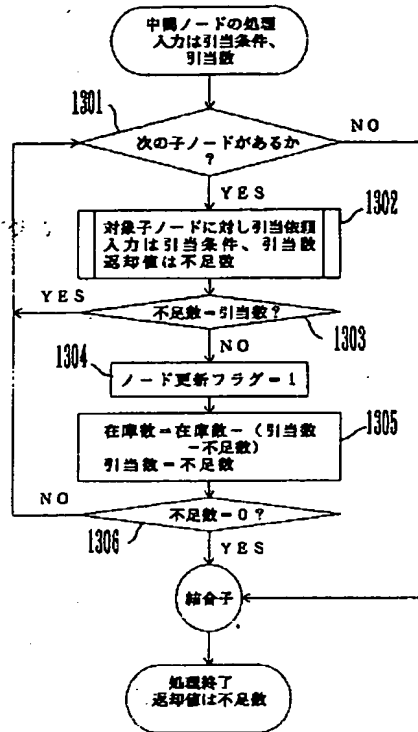
【図12】

引当依頼処理の動作フローチャート

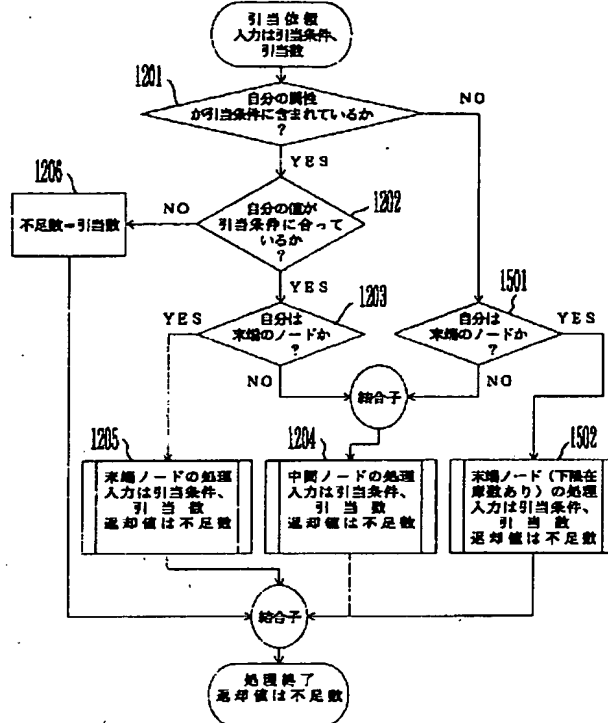


【図13】

中間ノードの処理の動作フローチャート

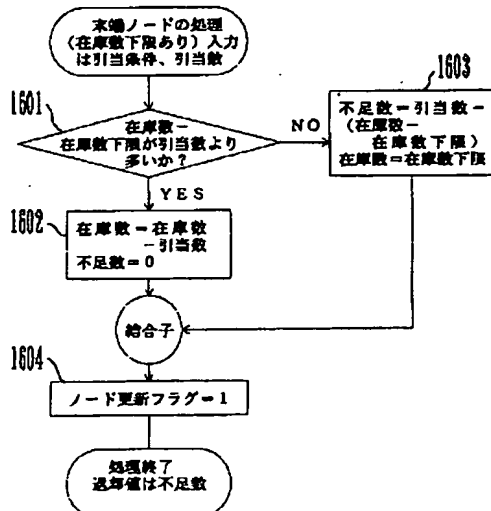


【図15】



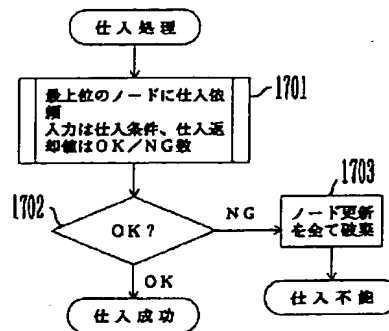
【図16】

在庫数下限がある末端ノードの処理の動作フローチャート

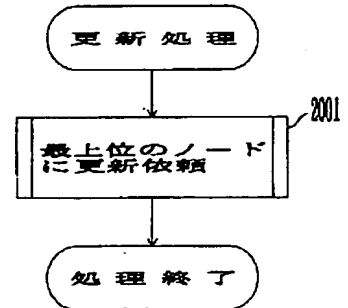


【図17】

仕入処理のメイン動作フローチャート



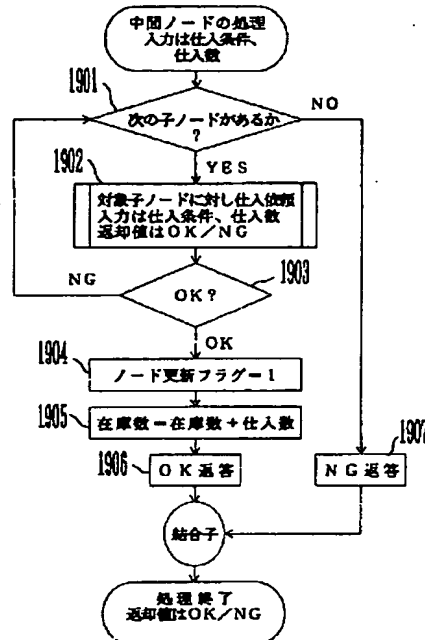
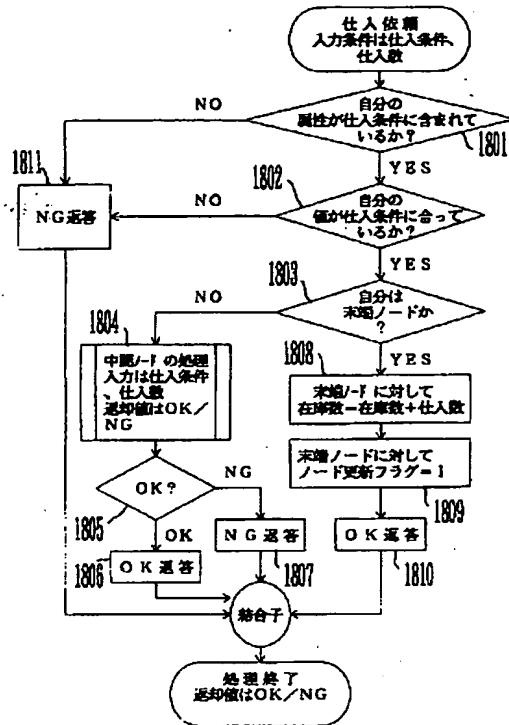
【図20】



【図18】

【図19】

仕入依頼処理の動作フローチャート 仕入処理における中間ノードの処理の動作フローチャート

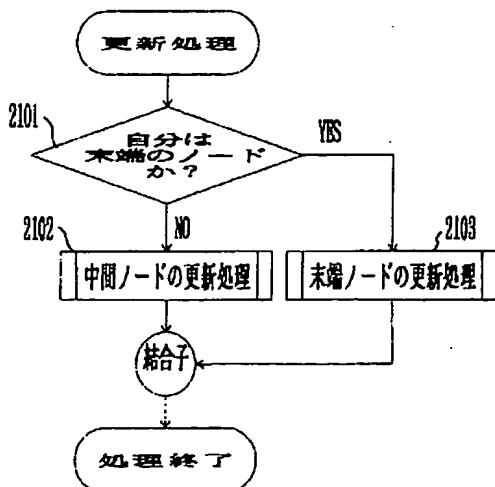


【図23】

更新処理における末端ノードの処理の動作フローチャート

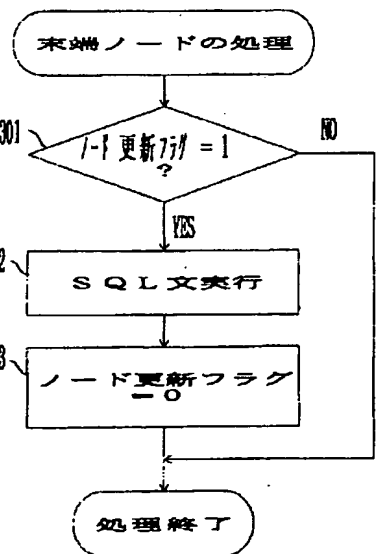
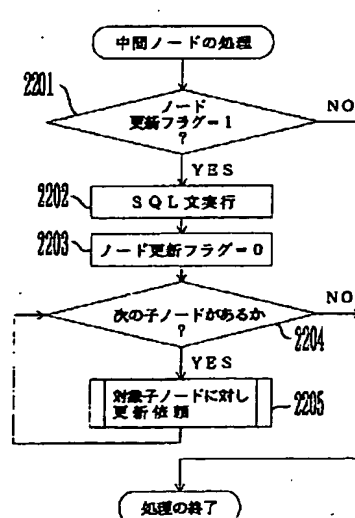
【図21】

更新依頼処理の動作フローチャート



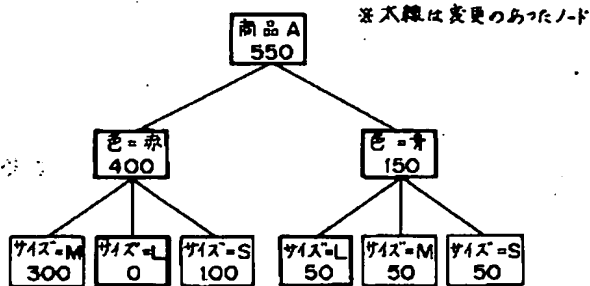
【図22】

更新処理における中間ノードの処理の動作フローチャート



【図24】

更新処理の動作説明図



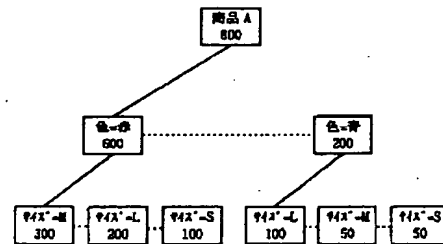
【図25】

在庫情報ツリーDBの他の実施の形態の構成図

(a)

ID	属性	値	在庫数	値ID	前ノードID	商品コード
1	商品	商品 A	800	0		商品 A
2	商品	商品 B	1200	0		商品 B
3	色	赤	600	1		商品 A
4	色	青	200		3	商品 A
5	サイズ	M	300	3		商品 A
6	サイズ	L	200		5	商品 A
7	サイズ	S	100		6	商品 A
8	サイズ	L	100	4		商品 A
9	サイズ	M	50		8	商品 A
10	サイズ	S	50		9	商品 A
11	サイズ	M	1000	2		商品 B
12	サイズ	S	200		11	商品 B
13	サイズ	L	200		12	商品 B
14	色	赤	400	11		商品 B
15	色	青	400		14	商品 B
16	色	黄	200		15	商品 B

(b)



【図26】

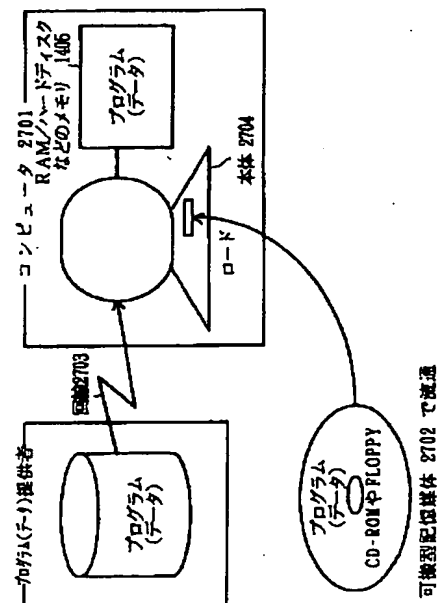
在庫情報ツリーDBの他の実施の形態の説明図

ID	属性	値	在庫数	値ID	前ノードID	商品コード
1	商品	商品 A	900	0		商品 A
2	商品	商品 B	1200	0		商品 B
3	色	赤	600	1		商品 A
4	色	青	200		17	商品 A
8	サイズ	M	800	3		商品 A
...中略...						
15	色	青	400		14	商品 B
16	色	黄	200		15	商品 B
17	色	白	100		3	商品 A

※斜字体が変更、追加された情報。

【図27】

本実施の形態を実現するプログラムが記録された記録媒体の説明図



フロントページの続き

(72)発明者 山本 里枝子
神奈川県川崎市中原区上小田中 4 丁目 1 番
1 号 富士通株式会社内

(72)発明者 桜井 浩
神奈川県川崎市中原区上小田中 4 丁目 1 番
1 号 富士通株式会社内